Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently amended) A method for balancing a workload for a plurality of processors in a multiple processor computer system, the system designed for processing a plurality of packets from a plurality of connections, the method comprising:

assigning a packet to a hash bucket determined by performing a predetermined hash function; and

queuing the hash bucket to a processor so that the workload of all the processors are balanced, wherein the queuing includes applying a queuing model to packets in the hash bucket to prevent packets from a particular connection from utilizing an excessive amount of the processor's time;

wherein the hash function relates to a predetermined number of hash buckets, and wherein a plurality of packets from different connections can be assigned to the same hash bucket.

2. (Original) The method of claim 1 wherein the step of assigning includes a step of using a source address, source port, destination address, and destination port parameters from the header section of the packet to identify the hash bucket.

3. (Original) The method of claim 1 wherein the hash function is designed to assign a plurality of packets from a connection to the same hash bucket.

4. (Original) The method of claim 1 further comprising monitoring the workload of each processor involved in the system.

5. (Currently amended) ~~The method of claim 1 wherein the step of queuing further includes:~~ A method for balancing a workload for a plurality of processors in a multiple processor computer system, the system designed for processing a plurality of packets from a plurality of connections, the method comprising:

assigning a packet to a hash bucket determined by performing a predetermined hash function; and

queuing the hash bucket to a processor so that the workload of all the processors are balanced, wherein the queuing includes

identifying statistically a percentage of capacity usage for each processor;

obtaining an imbalance distribution index for each processor from the identified percentage of capacity usage; and

distributing the hash buckets in proportion to the imbalance distribution index and in reverse proportion to a total imbalance index, wherein the total imbalance index is a summation of all obtained imbalance distribution indexes,

wherein the hash function relates to a predetermined number of hash buckets, and wherein a plurality of packets from different connections can be assigned to the same hash bucket.

6. (Canceled)

7. (Currently amended) A computer program for balancing workload for a plurality of processors in a multiple processor computer system, the system designed for processing a plurality of packets from a plurality of connections, the computer program comprising:

means for calculating a number of hash buckets to assign to a processor based on a ratio of an imbalance index of the processor and a total imbalance index, wherein the processor's imbalance index is based on a utilization rate of the processor and the system, and wherein the total imbalance index is based on the imbalance index of each processor;

means for assigning a packet to a hash bucket determined by performing a predetermined hash function; and

means for queuing the hash bucket to a processor so that the workload of all the processors are balanced,

wherein the hash function has a predetermined number of hash buckets, and wherein a plurality of packets from different connections can be assigned to the same hash bucket.

8. (Original) The computer program of claim 7 wherein the means for assigning includes means for using a source address, source port, destination address, and destination port parameters from the header section of the packet to identify the hash bucket.

9. (Original) The computer program of claim 7 wherein the hash function is designed to assign a plurality of packets from a connection to the same hash bucket.

10. (Original) The computer program of claim 7 further comprising means for monitoring the workload of each processor involved in the system.

11. (Currently amended) ~~The computer program of claim 7 wherein the means for queuing further includes means for:~~ A computer program for balancing workload for a plurality of processors in a multiple processor computer system, the system designed for processing a plurality of packets from a plurality of connections, the computer program comprising:

means for assigning a packet to a hash bucket determined by performing a predetermined hash function; and

means for queuing the hash bucket to a processor so that the workload of all the processors are balanced, wherein the means for queuing includes

identifying statistically a percentage of capacity usage for each processor;

obtaining an imbalance distribution index for each processor from the identified percentage of capacity usage; and

distributing the hash buckets in proportion to the imbalance distribution index and in reverse proportion to a total imbalance index, wherein the total imbalance index is a summation of all obtained imbalance distribution indexes,

wherein the hash function has a predetermined number of hash buckets, and wherein a plurality of packets from different connections can be assigned to the same hash bucket.

12. (Original) The computer program of claim 7 wherein the means for queuing further includes means for applying a queuing model for packets in the hash bucket to assure that packets from a particular connection do not unduly occupy the processor for an undesirable time period.

13. (Currently amended) A method for processing a plurality of connections with a plurality of timer threads by a plurality of computer processors in a multiple processor computer system, the method comprising:

providing a plurality of hash buckets related to a hash function;

mapping a connection to one of the hash buckets; and

assigning one or more ~~each~~ hash buckets to a processor timer thread based on a workload thereof so that the processor only processes the connection mapped to the assigned hash buckets,

wherein a plurality of timer threads for the computer processors thus process a plurality of connections simultaneously.

14. (Original) The method of claim 13 further comprising monitoring the workload of the each processor.

15. (Original) The method of claim 13 wherein the step of mapping is based on four connection parameters, which are local address, local port, remote address, and remote port.

16. (Currently amended) A computer program for processing a plurality of connections with a plurality of timer threads by a plurality of computer processors in a multiple processor computer system, the program comprising:

a plurality of hash buckets related to a predetermined hash function;

means for mapping a connection to one of the hash buckets; and

means for assigning a plurality of ~~each~~ hash buckets to a processor timer thread based on a workload thereof so that the processor only processes the connection mapped to the assigned hash buckets,

wherein a plurality of timer threads for the computer processors thus process a plurality of connections simultaneously.

17. (Original) The program of claim 16 further comprising means for monitoring the workload of the each processor.

18. (Original) The program of claim 16 wherein the means for mapping is based on four connection parameters, which are local address, local port, remote address, and remote port.